UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/081,079	02/21/2002	Rolf Gunter Erich Stegelmann	10157	2341

26890 7590 07/31/2006

JAMES M. STOVER
NCR CORPORATION
1700 SOUTH PATTERSON BLVD, WHQ4
DAYTON, OH 45479

EXAMINER	
WASSUM, LUKE S	

ART UNIT	PAPER NUMBER
2167	

DATE MAILED: 07/31/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/081,079	STEGELMANN ET AL.	
	Examiner	Art Unit	
	Luke S. Wassum	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 January 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-29* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *19 June 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 26 January 2006 has been entered.

Response to Request for Reconsideration

2. The Applicants' Request for Reconsideration, filed after final rejection on 10 November 2005, has been received, entered into the record, and considered.

3. The Applicants have submitted two declarations under 37 C.F.R. § 1.131, as well as supporting evidence. There were no amendments to the claims. Claims 1-29 remain pending in the application.

The Invention

4. The claimed invention is a database system including components for executing stored procedures that include at least one of a conditional expression, assignment expression, and dynamic database query language (e.g., SQL) statement, wherein low-level code is generated for each such expression, and inserted into the object code of the corresponding stored procedure.

Declaration under 37 C.F.R. § 1.131

5. The affidavit filed on 5 April 2005 under 37 CFR § 1.131 has been considered but is ineffective to overcome the **Oracle** reference ("PL/SQL User's Guide and Reference").

Formal Requirements of a Declaration

6. From MPEP § 715.04[R-2]:

> The following parties may make an affidavit or declaration under 37 CFR § 1.131:
>
> (A) All the inventors of the subject matter claimed.
>
> (B) An affidavit or declaration by less than all named inventors of an application is accepted where it is shown that less

than all named inventors of an application invented the subject matter of the claim or claims under rejection. For example, one of two joint inventors is accepted where it is shown that one of the joint inventors is the sole inventor of the claim or claims under rejection.

(C) **> If a petition under 37 CFR § 1.47 was granted or the application was accepted under 37 CFR § 1.42 or 1.43, the affidavit or declaration may be signed by the 37 CFR § 1.47 applicant or the legal representative, where appropriate.< .

(D) The assignee or other party in interest when it is not possible to produce the affidavit or declaration of the inventor. *Ex parte Foster*, 1903 C.D. 213, 105 O.G. 261 (Comm'r Pat. 1903).

Affidavits or declarations to overcome a rejection of a claim or claims must be made by the inventor or inventors of the subject matter of the rejected claim(s), a party qualified under 37 CFR §§ 1.42, 1.43, or 1.47, or the assignee or other party in interest when it is not possible to produce the affidavit or declaration of the inventor(s). Thus, where all of the named inventors of a pending application are not inventors of every claim of the application, any affidavit under 37 CFR § 1.131 could be signed by only the inventor(s) of the subject matter of the rejected claims. Further, where it is shown that a joint inventor is deceased, refuses to sign, or is otherwise unavailable, the signatures of the remaining joint inventors are sufficient. However, the affidavit or declaration, even though signed by fewer than all the joint inventors, must show completion of the invention by all of the joint inventors of the subject matter of the claim(s) under rejection. *In re Carlson*, 79 F.2d 900, 27 USPQ 400 (CCPA 1935).

An affidavit is a statement in writing made under oath before a notary public, magistrate, or officer authorized to administer oaths. See MPEP § 604 through § 604.06 for additional information regarding formal requirements of affidavits. 37 CFR § 1.68 permits a declaration to be used instead of an affidavit. The declaration must include an acknowledgment by the declarant that

willful false statements and the like are punishable by fine or imprisonment, or both (18 U.S.C. § 1001) and may jeopardize the validity of the application or any patent issuing thereon. The declarant must set forth in the body of the declaration that all statements made of the declarant's own knowledge are true and that all statements made on information and belief are believed to be true.

7. From MPEP 602 (II):

U.S. Patent and Trademark Office personnel are authorized to accept a statutory declaration under 28 U.S.C. § 1746 filed in the U.S. Patent and Trademark Office in lieu of an "oath" or declaration under 35 U.S.C. § 25 and 37 CFR § 1.68, provided that the statutory declaration otherwise complies with the requirements of law. Section 1746 of Title 28 of the United States Code provides:

> Whenever, under any law of the United States or under any rule, regulation, order, or requirement made pursuant to law, any matter is required to be supported, evidenced, established, or proved by sworn declaration, verification, certificate, statement, oath or affidavit, in writing of the person making the same (other than a deposition, or an oath of office, or an oath required to be taken before a specified official other than notary public), such matter may, with like force and effect, be supported, evidenced, established, or proved by the unsworn declaration, certificate, verification, or statement, in writing of such person which is subscribed by him, as true under penalty of perjury, and dated, in substantially the following form:
>
> [1] If executed without the United States:

"I declare (or certify, verify, or state) under penalty of perjury under the laws of the United States of America that the foregoing is true and correct. Executed on (date).
(Signature)."

[2] If executed within the United States its territories, possessions, or commonwealths:
"I declare (or certify, verify, or state) under penalty of perjury that the foregoing is true and correct. Executed on (date).
(Signature)."

8. The two submitted declarations fulfill all of the formal requirements for submission of a declaration under 37 C.F.R. §1.131. The examiner will now consider the merits of the declarations.

Allegation of FACTS

9. MPEP § 715.07 (I) states, *inter alia,*

The essential thing to be shown under 37 CFR § 1.131 is priority of invention and this may be done by any satisfactory evidence of the fact. FACTS, not conclusions, must be alleged. Evidence in the form of exhibits may accompany the affidavit or declaration. Each exhibit relied upon should be specifically referred to in the affidavit or declaration, in terms of what it is relied upon to show.

> A general allegation that the invention was completed prior to the date of the reference is not sufficient. *Ex parte Saunders*, 1883 C.D. 23, 23 O.G. 1224 (Comm'r Pat. 1883). Similarly, a declaration by the inventor to the effect that his or her invention was conceived or reduced to practice prior to the reference date, without a statement of facts demonstrating the correctness of this conclusion, is insufficient to satisfy 37 CFR § 1.131.
>
> The affidavit or declaration and exhibits must clearly explain which facts or data applicant is relying on to show completion of his or her invention prior to the particular date. Vague and general statements in broad terms about what the exhibits describe along with a general assertion that the exhibits describe a reduction to practice "amounts essentially to mere pleading, unsupported by proof or a showing of facts" and, thus, does not satisfy the requirements of 37 CFR § 1.131(b). *In re Borkowski*, 505 F.2d 713, 184 USPQ 29 (CCPA 1974). Applicant must give a clear explanation of the exhibits pointing out exactly what facts are established and relied on by applicant. 505 F.2d at 718-19, 184 USPQ at 33. See also *In re Harry*, 333 F.2d 920, 142 USPQ 164 (CCPA 1964) (Affidavit "asserts that facts exist but does not tell what they are or when the occurred.").

10. In the case of the instant declarations, the Applicants have alleged conclusions, not facts as is required under 37 C.F.R. § 1.131.

The declarations allege that the claimed invention was reduced to practice on or before 4 January 2001, a conclusion that has yet to be drawn based upon the submitted evidence.

A proper declaration is required to allege FACTS, which are fully supported by evidence.

The evidence submitted in support of the Applicants' declarations includes an **NCR** document titled "Performance Characterization for Teradata RDBMS Stored Procedures (TDSP) Feature in V2R4.1" dated 4 January 2001. One fact that could be alleged based upon this document is that (for instance) performance testing for some pre-release of the V2R4.1 version of NCR's Teradata RDBMS took place prior to 4 January 2001. This FACT would be fully supported by the document, since the document summarizes the test environment and results of the performance testing.

This evidence, however, by itself, fails to support the conclusion alleged in the Applicants' declarations that the claimed invention was reduced to practice prior to 4 January 2001.

As such, the Applicants have failed to meet their burden under 37 C.F.R. § 1.131(b).

Nevertheless, in order to advance prosecution of the application, the examiner will proceed to consider the remaining merits of the declaration and supporting evidence that has been submitted.

Conception

11. From MPEP § 715.07 (III):

> The affidavit or declaration must state FACTS and produce such documentary evidence and exhibits in support thereof as are available to show conception and completion of invention in this country or in a NAFTA or WTO member country (MPEP § 715.07(c)), at least the conception being at a date prior to the effective date of the reference. Where there has not been reduction to practice prior to the date of the reference, the applicant or patent owner must also show diligence in the completion of his or her invention from a time just prior to the date of the reference continuously up to the date of an actual reduction to practice or up to the date of filing his or her application (filing constitutes a constructive reduction to practice, 37 CFR § 1.131). As discussed above, 37 CFR § 1.131(b) provides three ways in which an applicant can establish prior invention of the claimed subject matter. The showing of facts must be sufficient to show:
>
> (A) reduction to practice of the invention prior to the effective date of the reference; or
>
> (B) conception of the invention prior to the effective date of the reference coupled with due diligence from prior to the reference date to a subsequent (actual) reduction to practice; or
>
> (C) conception of the invention prior to the effective date of the reference coupled with due diligence from prior to the reference date to the filing date of the application (constructive reduction to practice).

Conception is the mental part of the inventive act, but it must be capable of proof, as by drawings, complete disclosure to another person, etc. In *Mergenthaler v. Scudder*, 1897 C.D. 724, 81 O.G. 1417 (D.C. Cir. 1897), it was established that conception is

more than a mere vague idea of how to solve a problem; the means themselves and their interaction must be comprehended also.

12. From MPEP § 2138.04[R-1]:

> Conception has been defined as "the complete performance of the mental part of the inventive act" and it is "the formation in the mind of the inventor of a definite and permanent idea of the complete and operative invention as it is thereafter to be applied in practice...." *Townsend v. Smith*, 36 F.2d 292, 295, 4 USPQ 269, 271 (CCPA 1930). "[C]onception is established when the invention is made sufficiently clear to enable one skilled in the art to reduce it to practice without the exercise of extensive experimentation or the exercise of inventive skill." *Hiatt v. Ziegler*, 179 USPQ 757, 763 (Bd. Pat. Inter.1973). Conception has also been defined as a disclosure of an invention which enables one skilled in the art to reduce the invention to a practical form without "exercise of the inventive faculty." *Gunter v. Stream*, 573 F.2d 77, 197 USPQ 482 (CCPA 1978). See also *Coleman v. Dines*, 754 F.2d 353, 224 USPQ 857 (Fed. Cir. 1985) (It is settled that in establishing conception a party must show possession of every feature recited in the count, and that every limitation of the count must have been known to the inventor at the time of the alleged conception. Conception must be proved by corroborating evidence.)

13. In the case of the instant affidavit, since the Applicant is alleging actual reduction to practice before the date of the applied references, the date of conception is not an issue.

Diligence

14. From MPEP § 715.07(a):

> Where conception occurs prior to the date of the reference, but reduction to practice is afterward, it is not enough merely to allege that applicant or patent owner had been diligent. *Ex parte Hunter*, 1889 C.D. 218, 49 O.G. 733 (Comm'r Pat. 1889). Rather, applicant must show evidence of facts establishing diligence.
>
> What is meant by diligence is brought out in *Christie v. Seybold*, 1893 C.D. 515, 64 O.G. 1650 (6th Cir. 1893). In patent law, an inventor is either diligent at a given time or he is not diligent; there are no degrees of diligence. An applicant may be diligent within the meaning of the patent law when he or she is doing nothing, if his or her lack of activity is excused. Note, however, that the record must set forth an explanation or excuse for the inactivity; the USPTO or courts will not speculate on possible explanations for delay or inactivity. See *In re Nelson*, 420 F.2d 1079, 164 USPQ 458 (CCPA 1970). Diligence must be judged on the basis of the particular facts in each case. See MPEP § 2138.06 for a detailed discussion of the diligence requirement for proving prior invention.
>
> Under 37 CFR 1.131, the critical period in which diligence must be shown begins just prior to the effective date of the reference or activity and ends with the date of a reduction to practice, either actual or constructive (i.e., filing a United States patent application). Note, therefore, that only diligence before reduction to practice is a material consideration. The "lapse of time between the completion or reduction to practice of an invention and the filing of an application thereon" is not relevant to an affidavit or declaration under 37 CFR 1.131. See *Ex parte Merz*, 75 USPQ 296 (Bd. App. 1947).

15. As stated above, since the Applicant alleges an actual reduction to practice prior to the date of the applied reference, diligence is not at issue.

Reduction to Practice

16. Regarding reduction to practice, MPEP § 715.07 states:

> In general, proof of actual reduction to practice requires a showing that the apparatus actually existed and worked for its intended purpose.

17. From MPEP § 2138.05:

> Reduction to practice may be an actual reduction or a constructive reduction to practice which occurs when a patent application on the claimed invention is filed. The filing of a patent application serves as conception and constructive reduction to practice of the subject matter described in the application. Thus the inventor need not provide evidence of either conception or actual reduction to practice when relying on the content of the patent application. *Hyatt v. Boone*, 146 F.3d 1348, 1352, 47 USPQ2d 1128, 1130 (Fed. Cir. 1998).
>
> When a party to an interference seeks the benefit of an earlier-filed U.S. patent application, the earlier application must meet the requirements of 35 U.S.C. § 120 and 35 U.S.C. § 112, first paragraph for the subject matter of the count. The earlier

application must meet the enablement requirement and must contain a written description of the subject matter of the interference count. *Hyatt v. Boone*, 146 F.3d 1348, 1352, 47 USPQ2d 1128, 1130 (Fed. Cir. 1998). Proof of a constructive reduction to practice requires sufficient disclosure under the "how to use" and "how to make" requirements of 35 U.S.C. § 112, first paragraph. *Kawai v. Metlesics*, 480 F.2d 880, 886, 178 USPQ 158, 163 (CCPA 1973) (A constructive reduction to practice is not proven unless the specification discloses a practical utility where one would not be obvious. Prior art which disclosed an anticonvulsant compound which differed from the claimed compound only in the absence of a -CH2- group connecting two functional groups was not sufficient to establish utility of the claimed compound because the compounds were not so closely related that they could be presumed to have the same utility.). The purpose of the written description requirement is "to ensure that the inventor had possession, as of the filing date of the application relied on, of the specific subject matter later claimed by him." *In re Edwards*, 568 F.2d 1349, 1351-52, 196 USPQ 465, 467 (CCPA 1978). The written description must include all of the limitations of the interference count, or the applicant must show that any absent text is necessarily comprehended in the description provided and would have been so understood at the time the patent application was filed. Furthermore, the written description must be sufficient, when the entire specification is considered, such that the "necessary and only reasonable construction" that would be given it by a person skilled in the art is one that clearly supports each positive limitation in the count. *Hyatt v. Boone*, 146 F.3d at 1354-55, 47 USPQ2d at 1130-1132 (Fed. Cir. 1998) (The claim could be read as describing subject matter other than that of the count and thus did not establish that the applicant was in possession of the invention of the count.). See also *Bigham v. Godtfredsen*, 857 F.2d 1415, 1417, 8 USPQ2d 1266, 1268 (Fed. Cir. 1988) ("[t]he generic term halogen comprehends a limited number of species, and ordinarily constitutes a sufficient written description of the common halogen species, except where the halogen species are patentably distinct).

"In an interference proceeding, a party seeking to establish an actual reduction to practice must satisfy a two-prong test: (1) the party constructed an embodiment or performed a process that met every element of the interference count, and (2) the embodiment or process operated for its intended purpose." Eaton v. Evans, 204 F.3d 1094, 1097, 53 USPQ2d 1696, 1698 (Fed. Cir. 2000).

The same evidence sufficient for a constructive reduction to practice may be insufficient to establish an actual reduction to practice, which requires a showing of the invention in a physical or tangible form that shows every element of the count. Wetmore v. Quick, 536 F.2d 937, 942, 190 USPQ 223, 227 (CCPA 1976). For an actual reduction to practice, the invention must have been sufficiently tested to demonstrate that it will work for its intended purpose, but it need not be in a commercially satisfactory stage of development.

If a device is so simple, and its purpose and efficacy so obvious, construction alone is sufficient to demonstrate workability. King Instrument Corp. v. Otari Corp., 767 F.2d 853, 860, 226 USPQ 402, 407 (Fed. Cir. 1985).

For additional cases pertaining to the requirements necessary to establish actual reduction to practice see DSL Dynamic Sciences, Ltd. v. Union Switch & Signal, Inc., 928 F.2d 1122, 1126, 18 USPQ2d 1152, 1155 (Fed. Cir. 1991) ("events occurring after an alleged actual reduction to practice can call into question whether reduction to practice has in fact occurred"); Corona v. Dovan, 273 U.S. 692, 1928 C.D. 252 (1928) ("A process is reduced to practice when it is successfully performed. A machine is reduced to practice when it is assembled, adjusted and used. A manufacture [i.e., article of manufacture] is reduced to practice when it is completely manufactured. A composition of matter is reduced to practice when it is completely composed." 1928 C.D. at 262-263 (emphasis added).); Fitzgerald v. Arbib, 268 F.2d 763, 765-66, 122 USPQ 530, 531-32 (CCPA 1959) ("the reduction to practice of a three-dimensional design invention requires the production of an article embodying that design" in "other than a mere drawing").

> "The nature of testing which is required to establish a reduction to practice depends on the particular facts of each case, especially the nature of the invention." *Gellert v. Wanberg*, 495 F.2d 779, 783, 181 USPQ 648, 652 (CCPA 1974) ("an invention may be tested sufficiently ... where less than all of the conditions of actual use are duplicated by the tests"); *Wells v. Fremont*, 177 USPQ 22, 24-5 (Bd. Pat. Inter. 1972) ("even where tests are conducted under bench' or laboratory conditions, those conditions must fully duplicate each and every condition of actual use' or if they do not, then the evidence must establish a relationship between the subject matter, the test condition and the intended functional setting of the invention," but it is not required that all the conditions of all actual uses be duplicated, such as rain, snow, mud, dust and submersion in water).

18. In this case, an actual reduction to practice is alleged to have occurred prior to 4 January 2001. However, actual reduction to practice is not fully supported by the **NCR** document submitted as evidence.

The Applicants have failed to resolve (at least) the following issues which call into question the actual reduction to practice as alleged in the Applicants' declaration:

* Whether the performance testing described in the **NCR** document was performed under test conditions that represented actual conditions or realistically simulated conditions.

* Whether the performance testing described in the **NCR** document produced test results that demonstrated that the test was in fact successful.

* Whether the performance testing described in the **NCR** document was of such a nature as to establish that the software would work for its intended purpose.

* Whether the performance testing described in the **NCR** document was reproducible.

* Whether the performance testing described in the **NCR** document was performed on the final version of the software, or if there were further modifications to the software subsequent to the testing.

* Whether the performance testing described in the **NCR** document was performed on software that included all of the claim limitations, or if not, which claim limitations were included in the software tested.

19. In addition, the examiner points out that the evidence submitted by the Applicants in support of their declaration filed 7 April 2005 included a list of high level development milestones for version V2R4.1.0 of the Teradata RDBMS. This document contained evidence that integration testing of the software did not conclude until 9 March 2001, a fact which calls into question the Applicants' allegation in the instant declarations of an actual reduction to practice on or before 4 January 2001.

20. For the reasons cited above, the declarations filed by the Applicants under 37 C.F.R.§ 1.131 fail to establish that the claimed invention was reduced to practice prior to the critical period, and also fails to establish that the claimed invention was conceived prior to the critical period and diligently reduced to practice thereafter. As such, the affidavit is insufficient to establish invention prior to the prior art references relied upon in the rejections of record. The rejections are maintained by the examiner.

Claim Rejections - 35 USC § 103

21. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

22. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.

4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

23. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

24. Claims 1-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chow et al.** (U.S. Patent 5,875,334) in view of **Oracle** ("PL/SQL User's Guide and Reference").

25. Regarding claim 1, **Chow et al.** teaches a method of executing a stored procedure in a database system as claimed, the stored procedure containing at least an expression and a database query language statement, the method comprising:

a) identifying the expression in the stored procedure, the expression being according to one of plural predetermined types of expressions (see col. 8, lines 7-10);

b) generating low-level code representing the expression (see col. 8, lines 18-27); and

c) generating an object representing the stored procedure, the object containing the low-level code and one or more instructions representing the database query language statement (see col. 8, lines 27-32).

Chow et al. does not explicitly teach a method wherein the low-level code comprises object code.

Oracle, however, teaches a method wherein stored procedures, embodied in PL/SQL language routines, are compiled for native execution, requiring the generation of object code to represent the stored procedure (see pages 8-9 Compiling PL/SQL Code for Native Execution).

It would have been obvious to one of ordinary skill in the art at the time of the invention to compile stored procedures into object code, since execution of stored

procedures will be faster if they are compiled into native code (see pages 8-9 Compiling PL/SQL Code for Native Execution).

26. Regarding claim 15, **Chow et al.** teaches an article comprising at least one storage medium containing software as claimed, that when executed cause a database system to:

a) generate low-level code corresponding to a stored procedure having at least a first type expression and a second type expression, the first type expression selected from the group consisting of a conditional expression, an assignment expression and a dynamic database query language statement (see col. 8, lines 7-27);

b) create a predetermined type of code corresponding to the first type expression (see col. 8, lines 7-27);

c) provide the predetermined type of code in the low-level code to represent the first type expression (see col. 8, lines 7-27); and

d) provide one or more instructions representing the second type expression in the low-level code, the instructions different from the predetermined type of code (see col. 8, lines 27-32).

Chow et al. does not explicitly teach a method wherein the low-level code comprises object code.

Oracle, however, teaches a method wherein stored procedures, embodied in PL/SQL language routines, are compiled for native execution, requiring the generation of object code to represent the stored procedure (see pages 8-9 Compiling PL/SQL Code for Native Execution).

It would have been obvious to one of ordinary skill in the art at the time of the invention to compile stored procedures into object code, since execution of stored procedures will be faster if they are compiled into native code (see pages 8-9 Compiling PL/SQL Code for Native Execution).

27. Regarding claim 23, **Chow et al.** teaches a database system as claimed, comprising:

a) a plurality of nodes (see disclosure of client and server, col. 3, lines 15-19);

b) an evaluator module in a first one of the plurality of nodes (see disclosure that the stored procedure is invoked from the client, col. 4, lines 10-16);

c) an access module in a second one of the plurality of nodes, the access module to manage access to a portion of data stored in the database system (see disclosure that the statement is executed on the server, col. 3, lines 15-19); and

d) a controller in the first node adapted to execute a stored procedure low-level code, the low-level code containing a first type of code to represent an expression that is one of a conditional expression, an assignment expression and a dynamic statement, the low-level code containing a second, different type of code to represent a database query language statement (see col. 8, lines 7-27),

e) the controller adapted to submit the first type of code to the evaluator module to evaluate the expression (see col. 8, lines 7-27; see also disclosure of the separate PGM pointer for the 'procedural part' and the pointer to the executable plan for the query statement, col. 31, line 22 through col. 32, line 2); and

f) the controller adapted to submit a command corresponding to the database query language statement to the access module (see col. 3, lines 15-21; see

also disclosure of the separate PGM pointer for the 'procedural part' and the pointer to the executable plan for the query statement, col. 31, line 22 through col. 32, line 2).

Chow et al. does not explicitly teach a method wherein the low-level code comprises object code.

Oracle, however, teaches a method wherein stored procedures, embodied in PL/SQL language routines, are compiled for native execution, requiring the generation of object code to represent the stored procedure (see pages 8-9 Compiling PL/SQL Code for Native Execution).

It would have been obvious to one of ordinary skill in the art at the time of the invention to compile stored procedures into object code, since execution of stored procedures will be faster if they are compiled into native code (see pages 8-9 Compiling PL/SQL Code for Native Execution).

28. Regarding claim 27, **Chow et al.** teaches an article comprising at least one storage medium containing instructions for use in a database system as claimed, the instructions when executed causing the database system to:

a) access low-level code in response to invocation of a stored procedure, the low-level code containing first type of code representing an expression and a second type code representing a database query language statement (see col. 8, lines 7-27);

b) submit the first type code to an evaluator module to evaluate the expression (see col. 8, lines 7-27); and

c) submit a command corresponding to the database query language statement to an access module to access data specified by the database query language statement (see col. 3, lines 15-21).

Chow et al. does not explicitly teach a method wherein the low-level code comprises object code.

Oracle, however, teaches a method wherein stored procedures, embodied in PL/SQL language routines, are compiled for native execution, requiring the generation

of object code to represent the stored procedure (see pages 8-9 Compiling PL/SQL Code for Native Execution).

It would have been obvious to one of ordinary skill in the art at the time of the invention to compile stored procedures into object code, since execution of stored procedures will be faster if they are compiled into native code (see pages 8-9 Compiling PL/SQL Code for Native Execution).

29. Regarding claims 2 and 16, **Chow et al.** additionally teaches a method and article further comprising directly executing low-level code at run-time to evaluate the expression (see col. 27, lines 14-22).

30. Regarding claim 3, **Chow et al.** additionally teaches a method wherein directly executing the low-level code is performed in place of submitting a database query language statement to evaluate the expression (see col. 8, lines 7-27).

31. Regarding claim 4, **Chow et al.** additionally teaches a method wherein directly executing the low-level code consumes less database system resources than submitting a database query language statement to evaluate the expression (see col. 8, lines 42-48).

32. Regarding claims 5 and 18, **Chow et al.** additionally teaches a method and article further comprising:

a) submitting the low-level code to an evaluator module to evaluate the expression (see col. 8, lines 7-27); and

b) submitting a command corresponding to an access module in the database system to access data specified by the database query language statement (see col. 3, lines 15-21).

33. Regarding claim 6, **Chow et al.** additionally teaches a method wherein the database system has a first node containing a parsing engine and the evaluator module (see col. 8, lines 7-27) and a second node containing the access module (see disclosure that the statement is executed on the server, col. 3, lines 15-19), wherein submitting the command is performed by the parsing engine (see col. 8, lines 7-27).

34. Regarding claim 7, **Chow et al.** additionally teaches a method further comprising:

a) storing information pertaining to a variable and a constant used in the expression with the low-level code in the object (see col. 18, lines 25-60);

b) executing the low-level code during execution of the stored procedure using an evaluator module (see col. 27, lines 58-60); and

c) using the information pertaining to the variable and constant during execution of the low-level code to evaluate the expression (see col. 30, lines 27-52).

35. Regarding claim 8, **Chow et al.** additionally teaches a method wherein identifying the expression comprises identifying one of a conditional expression, an assignment expression and a dynamic database query language statement (see col. 8, lines 7-27).

36. Regarding claim 9, **Chow et al.** additionally teaches a method further comprising:

a) identifying a second expression in the stored procedure that is one of a conditional expression, an assignment expression, and a dynamic database query language expression (see col. 8, lines 7-27); and

b) generating second low-level code to represent the second expression, wherein generating the object comprises providing the second low-level code in the object (see col. 8, lines 7-27).

37. Regarding claim 10, **Chow et al.** additionally teaches a method wherein generating the object comprises generating the object containing the low-level code that is different from the instructions representing the database query language statement (see col. 7, lines 50-65).

38. Regarding claim 11, **Chow et al.** additionally teaches a method wherein generating the low-level code comprises generating assembly code (see disclosure that the procedural and non-procedural parts of the stored procedure are run through compilers, col. 7, line 66 through col. 8, line 6).

39. Regarding claim 12, **Chow et al.** additionally teaches a method further comprising:

a) storing the object in a predetermined location (see col. 4, lines 8-15); and

b) accessing the predetermined location to retrieve the object in response to invocation of the stored procedure (see col. 4, lines 8-15).

40. Regarding claim 13, **Chow et al.** additionally teaches a method wherein storing the object in the predetermined location comprises storing the object in a stored procedure table (see col. 4, lines 8-15).

41. Regarding claims 14 and 20, **Chow et al.** additionally teaches a method and article further comprising:

a) executing the object, wherein executing the object comprises submitting the low-level code to an evaluator module to execute the low-level code wherein the database query language statement is not one of the predetermined types of expressions (see disclosure of the separate PGM pointer for the 'procedural part' and the pointer to the executable plan for the query statement, col. 31, line 22 through col. 32, line 2); and

b) executing the one or more instructions representing the database query language statement without submitting the one or more instructions to the evaluator module (see disclosure of the separate PGM pointer for the 'procedural part' and the pointer to the executable plan for the query statement, col. 31, line 22 through col. 32, line 2).

42. Regarding claim 17, **Chow et al.** additionally teaches an article wherein the software when executed causes the database system to submit a database query language statement in the second type expression to the database system to evaluate the second type expression (see col. 3, lines 8-35).

43. Regarding claim 19, **Chow et al.** additionally teaches an article wherein the software when executed causes the database system to provide the predetermined type of code to the low-level code generator to add the low-level code (see col. 8, lines 7-27).

44. Regarding claims 21 and 25, **Chow et al.** additionally teaches a database system and article wherein the predetermined type of code corresponding to the first expression includes machine-level code (see disclosure that the procedural and non-procedural parts of the stored procedure are run through compilers, col. 7, line 66 through col. 8, line 6), and wherein the instructions representing the second type expression includes C code (see col. 1, lines 54-62).

45. Regarding claim 22, **Chow et al.** additionally teaches an article wherein the second type of expression comprises a Structured Query Language (SQL) statement (see SQL statements in the stored procedure listed at col. 4, lines 17-32).

46. Regarding claim 24, **Chow et al.** additionally teaches a database system wherein the controller comprises a parsing engine (see disclosure of parser at col. 8, lines 7-27).

47. Regarding claim 26, **Chow et al.** additionally teaches a database system wherein the first type code contains information identifying a type of the expression and a variable and constant used by the expression (see disclosure of variable bind-in, col. 5, line 49 through col. 6, line 36).

48. Regarding claim 28, **Chow et al.** additionally teaches an article wherein the instructions when executed cause the database system to generate the first type code for the expression being one of a conditional expression, assignment expression and dynamic statement, and provide the first type code in the low-level code (see col. 8, lines 7-27).

49. Regarding claim 29, **Chow et al.** additionally teaches an article wherein the instructions when executed cause the database system to generate the second type code which is different from the first type code, and provide the second type code in the low-

level code (see disclosure that there are procedural and non-procedural parts of the stored procedure, col. 7, line 66 through col. 8, line 6).

50. Claims 1-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chow et al.** (U.S. Patent 5,875,334) in view of **Gadhia et al.** ("PL/SQL Enhancements Planned for Future Oracle Releases").

51. Regarding claim 1, **Chow et al.** teaches a method of executing a stored procedure in a database system as claimed, the stored procedure containing at least an expression and a database query language statement, the method comprising:

a) identifying the expression in the stored procedure, the expression being according to one of plural predetermined types of expressions (see col. 8, lines 7-10);

b) generating low-level code representing the expression (see col. 8, lines 18-27); and

c) generating an object representing the stored procedure, the object containing the low-level code and one or more instructions representing the database query language statement (see col. 8, lines 27-32).

Chow et al. does not explicitly teach a method wherein the low-level code comprises object code.

Gadhia et al., however, teaches a method wherein stored procedures, embodied in PL/SQL language routines, are compiled for native execution, requiring the generation of object code to represent the stored procedure (see fourth top-level bullet under Planned PL/SQL Enhancements, titled Native Compilation of PL/SQL).

It would have been obvious to one of ordinary skill in the art at the time of the invention to compile stored procedures into object code, since execution of stored procedures will be faster if they are compiled into native code (see at least the second next-level bullet under the fourth top-level bullet, Facilitates faster execution of PL/SQL programs).

52. Regarding claim 15, **Chow et al.** teaches an article comprising at least one storage medium containing software as claimed, that when executed cause a database system to:

a) generate low-level code corresponding to a stored procedure having at least a first type expression and a second type expression, the first type expression selected from the group consisting of a conditional expression, an assignment expression and a dynamic database query language statement (see col. 8, lines 7-27);

b) create a predetermined type of code corresponding to the first type expression (see col. 8, lines 7-27);

c) provide the predetermined type of code in the low-level code to represent the first type expression (see col. 8, lines 7-27); and

d) provide one or more instructions representing the second type expression in the low-level code, the instructions different from the predetermined type of code (see col. 8, lines 27-32).

Chow et al. does not explicitly teach a method wherein the low-level code comprises object code.

Gadhia et al., however, teaches a method wherein stored procedures, embodied in PL/SQL language routines, are compiled for native execution, requiring the

generation of object code to represent the stored procedure (see fourth top-level bullet under Planned PL/SQL Enhancements, titled Native Compilation of PL/SQL).

It would have been obvious to one of ordinary skill in the art at the time of the invention to compile stored procedures into object code, since execution of stored procedures will be faster if they are compiled into native code (see at least the second next-level bullet under the fourth top-level bullet, Facilitates faster execution of PL/SQL programs).

53. Regarding claim 23, **Chow et al.** teaches a database system as claimed, comprising:

a) a plurality of nodes (see disclosure of client and server, col. 3, lines 15-19);

b) an evaluator module in a first one of the plurality of nodes (see disclosure that the stored procedure is invoked from the client, col. 4, lines 10-16);

c) an access module in a second one of the plurality of nodes, the access module to manage access to a portion of data stored in the database system (see disclosure that the statement is executed on the server, col. 3, lines 15-19); and

d) a controller in the first node adapted to execute a stored procedure low-level code, the low-level code containing a first type of code to represent an expression that is one of a conditional expression, an assignment expression and a dynamic statement, the low-level code containing a second, different type of code to represent a database query language statement (see col. 8, lines 7-27),

e) the controller adapted to submit the first type of code to the evaluator module to evaluate the expression (see col. 8, lines 7-27; see also disclosure of the separate PGM pointer for the 'procedural part' and the pointer to the executable plan for the query statement, col. 31, line 22 through col. 32, line 2); and

f) the controller adapted to submit a command corresponding to the database query language statement to the access module (see col. 3, lines 15-21; see also disclosure of the separate PGM pointer for the 'procedural part' and the pointer to the executable plan for the query statement, col. 31, line 22 through col. 32, line 2).

Chow et al. does not explicitly teach a method wherein the low-level code comprises object code.

Gadhia et al., however, teaches a method wherein stored procedures, embodied in PL/SQL language routines, are compiled for native execution, requiring the generation of object code to represent the stored procedure (see fourth top-level bullet under Planned PL/SQL Enhancements, titled Native Compilation of PL/SQL).

It would have been obvious to one of ordinary skill in the art at the time of the invention to compile stored procedures into object code, since execution of stored procedures will be faster if they are compiled into native code (see at least the second next-level bullet under the fourth top-level bullet, Facilitates faster execution of PL/SQL programs).

54. Regarding claim 27, **Chow et al.** teaches an article comprising at least one storage medium containing instructions for use in a database system as claimed, the instructions when executed causing the database system to:

a) access low-level code in response to invocation of a stored procedure, the low-level code containing first type of code representing an expression and a

second type code representing a database query language statement (see col. 8, lines 7-27);

b) submit the first type code to an evaluator module to evaluate the expression (see col. 8, lines 7-27); and

c) submit a command corresponding to the database query language statement to an access module to access data specified by the database query language statement (see col. 3, lines 15-21).

Chow et al. does not explicitly teach a method wherein the low-level code comprises object code.

Gadhia et al., however, teaches a method wherein stored procedures, embodied in PL/SQL language routines, are compiled for native execution, requiring the generation of object code to represent the stored procedure (see fourth top-level bullet under Planned PL/SQL Enhancements, titled Native Compilation of PL/SQL).

It would have been obvious to one of ordinary skill in the art at the time of the invention to compile stored procedures into object code, since execution of stored procedures will be faster if they are compiled into native code (see at least the second

next-level bullet under the fourth top-level bullet, Facilitates faster execution of PL/SQL programs).

55. Regarding claims 2 and 16, **Chow et al.** additionally teaches a method and article further comprising directly executing low-level code at run-time to evaluate the expression (see col. 27, lines 14-22).

56. Regarding claim 3, **Chow et al.** additionally teaches a method wherein directly executing the low-level code is performed in place of submitting a database query language statement to evaluate the expression (see col. 8, lines 7-27).

57. Regarding claim 4, **Chow et al.** additionally teaches a method wherein directly executing the low-level code consumes less database system resources than submitting a database query language statement to evaluate the expression (see col. 8, lines 42-48).

58. Regarding claims 5 and 18, **Chow et al.** additionally teaches a method and article further comprising:

a) submitting the low-level code to an evaluator module to evaluate the expression (see col. 8, lines 7-27); and

b) submitting a command corresponding to an access module in the database system to access data specified by the database query language statement (see col. 3, lines 15-21).

59. Regarding claim 6, **Chow et al.** additionally teaches a method wherein the database system has a first node containing a parsing engine and the evaluator module (see col. 8, lines 7-27) and a second node containing the access module (see disclosure that the statement is executed on the server, col. 3, lines 15-19), wherein submitting the command is performed by the parsing engine (see col. 8, lines 7-27).

60. Regarding claim 7, **Chow et al.** additionally teaches a method further comprising:

a) storing information pertaining to a variable and a constant used in the expression with the low-level code in the object (see col. 18, lines 25-60);

b) executing the low-level code during execution of the stored procedure using an evaluator module (see col. 27, lines 58-60); and

c) using the information pertaining to the variable and constant during execution of the low-level code to evaluate the expression (see col. 30, lines 27-52).

61. Regarding claim 8, **Chow et al.** additionally teaches a method wherein identifying the expression comprises identifying one of a conditional expression, an assignment expression and a dynamic database query language statement (see col. 8, lines 7-27).

62. Regarding claim 9, **Chow et al.** additionally teaches a method further comprising:

a) identifying a second expression in the stored procedure that is one of a conditional expression, an assignment expression, and a dynamic database query language expression (see col. 8, lines 7-27); and

b) generating second low-level code to represent the second expression, wherein generating the object comprises providing the second low-level code in the object (see col. 8, lines 7-27).

63. Regarding claim 10, **Chow et al.** additionally teaches a method wherein generating the object comprises generating the object containing the low-level code that

is different from the instructions representing the database query language statement (see col. 7, lines 50-65).

64. Regarding claim 11, **Chow et al.** additionally teaches a method wherein generating the low-level code comprises generating assembly code (see disclosure that the procedural and non-procedural parts of the stored procedure are run through compilers, col. 7, line 66 through col. 8, line 6).

65. Regarding claim 12, **Chow et al.** additionally teaches a method further comprising:

a) storing the object in a predetermined location (see col. 4, lines 8-15); and

b) accessing the predetermined location to retrieve the object in response to invocation of the stored procedure (see col. 4, lines 8-15).

66. Regarding claim 13, **Chow et al.** additionally teaches a method wherein storing the object in the predetermined location comprises storing the object in a stored procedure table (see col. 4, lines 8-15).

67. Regarding claims 14 and 20, **Chow et al.** additionally teaches a method and article further comprising:

a) executing the object, wherein executing the object comprises submitting the low-level code to an evaluator module to execute the low-level code wherein the database query language statement is not one of the predetermined types of expressions (see disclosure of the separate PGM pointer for the 'procedural part' and the pointer to the executable plan for the query statement, col. 31, line 22 through col. 32, line 2); and

b) executing the one or more instructions representing the database query language statement without submitting the one or more instructions to the evaluator module (see disclosure of the separate PGM pointer for the 'procedural part' and the pointer to the executable plan for the query statement, col. 31, line 22 through col. 32, line 2).

68. Regarding claim 17, **Chow et al.** additionally teaches an article wherein the software when executed causes the database system to submit a database query language statement in the second type expression to the database system to evaluate the second type expression (see col. 3, lines 8-35).

69. Regarding claim 19, **Chow et al.** additionally teaches an article wherein the software when executed causes the database system to provide the predetermined type of code to the low-level code generator to add the low-level code (see col. 8, lines 7-27).

70. Regarding claims 21 and 25, **Chow et al.** additionally teaches a database system and article wherein the predetermined type of code corresponding to the first expression includes machine-level code (see disclosure that the procedural and non-procedural parts of the stored procedure are run through compilers, col. 7, line 66 through col. 8, line 6), and wherein the instructions representing the second type expression includes C code (see col. 1, lines 54-62).

71. Regarding claim 22, **Chow et al.** additionally teaches an article wherein the second type of expression comprises a Structured Query Language (SQL) statement (see SQL statements in the stored procedure listed at col. 4, lines 17-32).

72. Regarding claim 24, **Chow et al.** additionally teaches a database system wherein the controller comprises a parsing engine (see disclosure of parser at col. 8, lines 7-27).

73. Regarding claim 26, **Chow et al.** additionally teaches a database system wherein the first type code contains information identifying a type of the expression and a variable and constant used by the expression (see disclosure of variable bind-in, col. 5, line 49 through col. 6, line 36).

74. Regarding claim 28, **Chow et al.** additionally teaches an article wherein the instructions when executed cause the database system to generate the first type code for the expression being one of a conditional expression, assignment expression and dynamic statement, and provide the first type code in the low-level code (see col. 8, lines 7-27).

75. Regarding claim 29, **Chow et al.** additionally teaches an article wherein the instructions when executed cause the database system to generate the second type code which is different from the first type code, and provide the second type code in the low-level code (see disclosure that there are procedural and non-procedural parts of the stored procedure, col. 7, line 66 through col. 8, line 6).

Response to Arguments

76. Applicant's arguments filed 10 November 2005 have been fully considered but they are not persuasive.

77. Regarding the Applicant's affidavit under U.S.C. § 1.131, the affidavit is ineffective to disqualify the prior art as alleged by the Applicant, for the reasons stated above.

Conclusion

78. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Rosenberg ("Bringing Java to the Enterprise: Oracle on its Java Server Strategy") teaches five aspects of Java product strategy included in the Oracle 8.1 software development kit.

Halfhill ("How to Soup Up Java") teaches ways to boost Java's performance.

Nobody Win ("Soory For 9i") is a listserv posting documenting some of the new features of the Oracle9i release, including support for native compilation and improved optimization of PL/SQL.

Oracle ("Oracle8i Java Developer's Guide, Release 3 (8.1.7)") teaches support for native compilation of certain Java classes.

Llewellyn et al. ("New PL/SQL Features in Action: Real World Case Studies") teaches two new features of Release 1 and Release 2 of Oracle9i: native compilation of PL.SQL and *index-by-varchar2* tables.

Quest Software ("Pipeline Newsletter Archive - Oracle PL/SQL") is an index of Quest Pipeline Newsletters on Oracle PL/SQL, and establishes the date of the Gadhia et al. reference as June 2000.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Luke S. Wassum whose telephone number is 571-272-4119. The examiner can normally be reached on Monday-Friday 8:30-5:30, alternate Fridays off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John R. Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

In addition, INFORMAL or DRAFT communications may be faxed directly to the examiner at 571-273-4119. Such communications must be clearly marked as INFORMAL, DRAFT or UNOFFICIAL.

Customer Service for Tech Center 2100 can be reached during regular business hours at (571) 272-2100, or fax (571) 273-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Luke S. Wassum
Primary Examiner
Art Unit 2167

lsw
25 July 2006